eKashu –

Frequently Asked Questions

Document addressing commonly raised support queries and issues for new integrators.

Issue: 1 (November 2013)

Author: Fred Spooner (Integration Support)

Action	Name	Date
Created	F. Spooner	13th November 2013

Release

Version	Date Released	Change Notice	Pages Affected	Remarks

Distribution List

Name	Organisation	Title

creditcall
card payment solutions

CreditCall Corporation
1133 Broadway, Suite 706
New York, NY 10010
USA

T +1 (212) 807 4979
F +1 (212) 330 8006

CreditCall Ltd
Merchants House South
Wapping Road
Bristol BS1 4RW

T 0117 930 4455
F 0117 930 4477

E support@creditcall.com
W www.creditcall.com

CONTENTS

General

Where can I find documentation for eKashu?

The latest version of the eKashu integration guide is available in PDF format on the CreditCall website:

http://www.creditcall.com/hres/ekashu%20paymentpage%20integration%20guide.pdf

Can I use real payment cards on the test platform?

From a security point of view, it is not considered good practice to use real payment cards in a testing environment, and accordingly, we do not recommend it. That said, as the test platform simulates acquirer responses and is not connected to any real acquirers, funds will not be reserved or charged against a live, enrolled payment card.

A quick web search will provide you with suitable lists of card data for testing purposes, and generators that produce Luhn-valid card numbers. Alternatively, contact ekashusupport@creditcall.com and we will be more than happy to provide you with a list of card details that can be used appropriately for testing, along with AVS (Address Verification System) and CVV (Card Verification Value) information that will allow you to fully test your application's validation of cardholder details, and handle bank declines and other errors accordingly.

Card details that are not registered on the test platform will pass AVS and CVV validation regardless of whether they are correct or not (i.e. any given address or CVV will be returned as "Matched"), so it is advisable to use a combination of test data provided by CreditCall as well as card data generated or sourced by yourself for the testing process.

It is worth noting that any test card data provided by CreditCall or found anywhere else will not work on any live platform. A card number that passes a Luhn algorithm check does not mean that it is enrolled with any bank or will work on a live payment platform. It is discouraged to ever use test card data on a live system.

How long does it take to switch from a test account to a live account?

Once we have received a signed merchant registration form we set up your account on our platform using the merchant account details provided by your acquirer. This can take up to 5 working days.

Once this stage is complete and we have successfully tested it we will provide you with the configuration information.

Keep in mind that it can take up to 2 weeks to register as a merchant with 3-D Secure and it is always advisable to allow as much time as possible for the switch over before going completely live.

Features & Functionality

Can I use eKashu to make repeat payments (without storing card details)?

eKashu is primarily intended to safely capture, authenticate, and process payment card information.

However, every successful card authorisation processed by eKashu will return `ekashu_card_hash` and `ekashu_card_reference` which together are commonly known as a "token". Tokens can be safely stored (e.g. in a database) without coming under scope for PCI DSS, and submitted to eKashu to make subsequent payments from a card, without requiring the card details themselves.

This can make your application's user experience more convenient for your customers, as rather than having to enter their card details every time they want to pay, the token that refers to their card details can be retrieved from your database and passed to eKashu to take payment. Rather than having to input the card number, expiry date, and Card Verification Value (CVV), you can set `ekashu_card_hash` and `ekashu_card_reference` as hidden elements in your form to the card hash and card reference associated with the customer. Other than the customer pressing 'pay' in order to submit payment, this method essentially negates the need for cardholder interaction in this situation.

If `ekashu_verification_value_verify` is set to "true" when performing a transaction in this way, eKashu will still prompt the cardholder for the CVV. This can be useful as an extra verification step to ensure that the user legitimately possesses the payment card in question, and reduces the risk of fraud and resultant chargebacks. As the card number isn't supplied in this case, the 3-D Secure process cannot take place and is bypassed automatically.

If more flexibility over transactions is required, it is worth investigating CardEaseXML, the SDK on which eKashu is based. In addition to accepting regular payment card information, tokenised card information returned by eKashu can be processed via CardEaseXML, and vice versa.

Additionally, CardEaseXML supports the setup and management of recurring payments. While CardEaseXML and eKashu complement each other to create a more powerful payment processing application, the use of CardEaseXML will increase the lead time required for integration and testing before going live.

For more information about CardEaseXML, visit http://www.creditcall.com/cardease_xml.html.

Remember: storing card information puts the merchant in scope for PCI DSS assessment and should be avoided. You should ONLY store the tokenised card hash and card reference returned by eKashu.

Is there any way to submit data to eKashu using HTTP GET instead of POST?

No. There are certain core parts of the payment page that require HTTPS POST, and GET exposes too much data that is at risk of being manipulated or edited, affecting the transaction. You can however, pass your own reference to eKashu as part of the HTML form if necessary. See the eKashu integration guide for more information.

You can specify GET data in your success/failure/callback URLs if you wish, but we strongly discourage including any sensitive data, or data that could influence your site's response to the data returned by eKashu.

How do I turn off 3-D Secure Verification?

In some cases, it is not necessary to make 3-D Secure verification mandatory, for example if eKashu is being used to process card data in a back-office environment. As 3-D Secure functionality ensures fraud liability shift as well as an extra layer of security, it is not recommended to disable it. That said, you are able to disable 3-D Secure verification by adding the hidden form element "ekashu_3d_secure_verify" and setting it to "false".

Remember: Maestro card scheme rules require 3-D Secure validation for e-commerce transactions. Failure to do so may result in fines from the acquiring bank, or automatic declining of Maestro transactions. As such, eKashu will override the option to bypass 3-D Secure when presented with a Maestro card.

How can I provide my customers with a transaction reference?

You can pass your own reference to eKashu using the form field `ekashu_reference`. This will be returned exactly as it was posted when you are redirected back to your site after payment via eKashu. Alternatively, if you prefer, you may use `ekashu_transaction_id` as a unique payment reference.

Does eKashu automatically email confirmation of payment to customers?

eKashu does not provide this functionality. If you wish to email customers confirmation of a transaction, you will need to implement this yourself on your website. Naturally, you can use data returned by eKashu (such as masked card number, transaction reference, etc.) to supplement the confirmation message.

We recommend that you investigate eKashu's callback functionality in the eKashu Integration Guide.

How do I test/use eKashu with PayPal?

In order to test eKashu with PayPal, you will require a PayPal developer account if you do not already have one. PayPal offers a 'sandbox' platform for test purposes, which allows the creation of test business/merchant accounts, and test personal/buyer accounts, without the need to supply information such as bank details, etc.

For an overview of PayPal's Sandbox testing environment, and instructions on how to create test accounts, visit https://developer.paypal.com/webapps/developer/docs/classic/lifecycle/ug_sandbox/#accounts.

You will require a test business account (normally this is automatically configured when you register), and at least one test buyer account so that you can perform a PayPal transaction via eKashu from start to finish.

Once your test credentials are configured, contact ekashusupport@creditcall.com with the following information:

- Your Terminal ID (ekashu seller ID)
- Your PayPal Sandbox Business account email

We will endeavour to configure your Terminal ID for PayPal testing within a working day.

Once the above steps have been followed, you can specify the `ekashu_payment_method` and `ekashu_payment_methods` fields, you will need to allow the CreditCall/eKashu API access to your Sandbox account. The process is essentially the same for live accounts, except that your PayPal Business account will be registered on the eKashu/CardEase platform during the boarding process with CreditCall. Naturally, the API credentials for a live account will differ as well. Refer to the eKashu integration guide or contact eKashu support for more information.

CUSTOMISATION

Can I turn off/change the address fields/card logos/other elements on the payment page?

Address fields can be disabled by setting: `ekashu_card_address_verify`, `ekashu_card_zip_code_verify`, `ekashu_card_address_required` to "`false`". More information on this can be found in the eKashu integration guide.

We discourage using CSS to hide elements on the page, as card scheme rules state that 3-D Secure logos and card scheme logos (including the space around them) should be present and intact on the payment page. Other than that, feel free to stylise the page how you wish.

On the test platform, eKashu will display all card scheme logos. When you go live, only the card schemes in which you are enrolled to take payment will be displayed.

Note: *Please bear in mind that if you hide an input field that may be required by certain cards (such as start date) we cannot be held responsible if errors occur when attempting to take payment.*

Is it possible to change which fields are mandatory on the eKashu page (such as CVV)?

We cannot change which fields are mandatory or not, as this depends entirely on the card scheme of the customer. If the customer doesn't enter a CVV when one is required, eKashu will then prompt them to enter the valid CVV once they've attempted to submit the form.

Can CreditCall host a custom eKashu stylesheet for me?

Yes, although we encourage you to host it on your own HTTPS server so that you can maintain it without needing to send us a new one every time it is updated.

NOTE: *all CSS/images/other elements that you are using to redecorate the eKashu payment page should be hosted securely (https), otherwise the user's browser will display warnings about mixed secure/unsecure content. This could cause lost revenue due to users mistaking the warnings as a potential fraud/phishing attempt.*

HASH KEYS/HASH CODES

How do I obtain a Hash Key for use with eKashu?

Email salesadmin@creditcall.com with the subject "hash key request", providing your **Seller ID** and **Transaction Key**. We will endeavour to respond with your **Hash Key** within 24 working hours.

Keep in mind that there will be one **Hash Key** provided for testing, and a separate one must be requested for live use, so please provide the relevant **Seller ID** and **Transaction Key**. Test **Seller IDs** commonly begin with a '9', and live IDs with a '2'.

How do I generate an `ekashu_hash_code`?

[This process is explained in the eKashu integration guide.]

In brief, `ekashu_hash_code` is a SHA1 hash generated from a concatenated string composed of the merchant's secret **Hash Key**, **Seller ID**, the **Transaction Reference**, and the transaction **Amount**. It is used to validate the transaction between being posted to eKashu from the merchant's site, and the user's browser being redirected, to ensure that the data has not been tampered with (i.e. altering the amount!)

The guide also contains sample code explaining how to generate the hash code in C# and in PHP, in a format that should be fairly simple to apply in a different programming language if required.

Note: You will need to request a hash key from CreditCall if you do not already have one. See *"How do I obtain a Hash Key for use with eKashu?"* below.

Take care when generating your hash code. If you are met with an error stating "invalid POST form field value for: ekashu_hash_code", double check the data that is being passed to the function. For example, when using Major AmountUnits, '$2' will not result in the same hash code as '$2.00', or even '2' or '2.00'. The same applies to the transaction reference. The strings that are passed to your function to create your hash code must be identical to those that are POSTed to eKashu.

What is the difference between ekashu_hash_code and ekashu_hash_code_result?

`ekashu_hash_code_result` is similar to `ekashu_hash_code` except that it validates the *response* from eKashu after a transaction has been processed. It is constructed from the merchant's **Hash Key**, **Seller ID**, **Transaction Reference**, and **Auth Result**.

Using the variables described above, a **Hash Code** can be generated by your own application that can be used to validate against the `ekashu_hash_code_result` returned by eKashu.

When produced correctly, the user-generated **Hash Code** will be an exact match. This will prevent a user tampering with the POST data, for example fooling your website into registering a transaction as successful when it had actually failed or been declined.

CALLBACKS

What is the difference between a success/failure URL and a success/failure *callback* URL?

The `ekashu_success_url` and `ekashu_failure_url` fields redirect the customer's browser back to the specified page upon a successful or failed transaction. The callback URLs (`ekashu_callback_success_url` and `ekashu_failure_url`) send data to a specified page 'behind the scenes', usually to execute a script (specified by the URL) that makes use of the data returned by eKashu when a transaction takes place.

If a callback fails, our servers will retry a callback after 1 minute, then 2 minutes, then 4 minutes, and so on, until it is successful (HTTP code 200), or up to a maximum of 3 days.

This process is intended to be used as a 'failsafe' should the customer's browser fail to redirect back to the success or failure page. A brief example of this feature would be a script that automatically closes a customer's order and emails them confirmation of the transaction once the callback has been performed.

If I specify a callback URL as well as a redirect URL, is the same information posted to both?

Yes, when the `ekashu_include_post` field is set to "true". See the eKashu integration guide for more information.

Why is my callback URL not doing anything even when a transaction is successfully processed by eKashu?

This issue often arises when the callbacks are directed to the URL of a server behind a firewall, for instance, an in-house development server. If you think that this is the case, please contact ekashusupport@creditcall.com and we will provide you with the IP addresses that need to be added as exceptions to your firewall's rules. Our firewalls do not support callbacks on any ports other than 443 and 80.

Also, keep in mind that local server IP addresses or 'localhost' will not work as the eKashu servers will not know where to direct the callback. Our servers are configured to reattempt failed callbacks after a set period of time, up to a maximum limit, in order to minimise a loss of information should there be a connection issue between our servers and your site.

Finally, your callback script should expect POST data to be returned by eKashu, although you can specify GET parameters in the callback URL if necessary.

Common Errors/Warnings

How can I force eKashu to fail/decline a transaction during testing?

The simplest way to do this is to set the transaction amount to greater than £130 (the maximum limit on the test platform) three times, or use one of our test cards with an incorrect CVV (Card Verification Value).

If a payment is declined, is it possible to inform the user why?

Our payment page performs routine validation checks of your customer's card details, to ensure the card number is valid (Luhn check), that the expiry date is in the future, and so on.

The acquiring bank only responds to validated card information with either "AUTH CODE: xxxxxx" or "DECLINED". It does not provide a reason as to why the card was declined.

If a payment fails due to a failure to validate the card details (incorrect format, communication failure, etc.), eKashu will return the relevant error message.

eKashu returns the error: "Invalid POST form field value for: `ekashu_seller_id`" and/or "Invalid POST form field value for: `ekashu_seller_key`".

Remember that your `ekashu_seller_key` is the *first eight characters* of the **Transaction Key** provided to you upon registration to Test WebMIS (on our test platform), or by our Sales Admin team during the boarding process on our live platform. Your `ekashu_seller_id` is the **Terminal ID** provided in the same way (registration on Test WebMIS or via our Sales Admin team for live boarding).

If you haven't signed up for a Test WebMIS account, then you can do so at: https://testwebmis.creditcall.com/registration.php

If you have registered on Test WebMIS but don't know your Terminal ID and Transaction Key, first check your email inbox/spam folder for an email from noreply@creditcall.com and see if the initial registration email is there. If not, contact ekashusupport@creditcall.com.

Remember: *There are two different pairs of Terminal ID and Transaction Key for each platform: test, and live. They are not interchangeable between the test and live platforms.*

To obtain live credentials, you will need to contact the CreditCall Sales Admin team about our boarding procedures. In most cases, live Terminal IDs start with 2, and test Terminal IDs start with 9.

eKashu responds with the error: "The seller cannot accept a payment this high/low".

On the eKashu **test** platform, there is a maximum transaction amount of 130 major units (e.g. $130, £130, €130 etc.). The minimum transaction amount is 0.25 major units (e.g. 25¢, 25p etc.).

On the **live** platform, these limits can be configured to whatever you like when setting up your merchant account. The default limit is between 0.50 major units and 100.00 major units. If you need your minimum/maximum limits altered, please contact salesadmin@creditcall.com.

***Please note**: we cannot change the transaction amount limit on the test platform, as it can cause other integrator's tests to fail unexpectedly.*

During redirects on the eKashu test platform, I get a security warning that data will be sent over an unencrypted connection/I get an invalid certificate warning.

On the test platform the 3-D Secure Access Control Server (ACS) will present the cardholder's browser with a certificate that was not issued by a trusted certificate authority.

This is intentional as it helps to make sure that the ACS is not mistakenly used for live transactions. When using the test platform the certificate warning displayed by the browser relating to the ACS can be safely ignored allowing authentication to continue.

It has been noted that a similar warning tends to occur on the live platform when using Mozilla Firefox; however it can occur in other browsers as well. The most common reason for this is that a custom CSS file, and/or other HTML elements such as images, used by the merchant are not hosted on a secure (HTTPS) server.

Ensure that your success and failure redirect URLs also use HTTPS.

If you are using the live platform, you are certain that the criteria described above have been met, and your users are met with the same warning, it is inadvisable to continue entering sensitive data, and you should contact ekashusupport@creditcall.com.